Implicit Newton-Krylov Methods for Modeling Blast Furnace Stoves*

J. W. Howse G. A. Hansen
D. J. Cagliostro
Computational Science Methods Group, X-CM
P.O. Box 1663, Mail Stop F645
Los Alamos National Laboratory
Los Alamos, NM 87545

K. R. Muske
Department of Chemical Engineering
Villanova University
800 Lancaster Ave.
Villanova, PA 19085

Abstract

In this paper we discuss the use of an implicit Newton-Krylov method to solve a set of partial differential equations representing a physical model of a blast furnace stove. The blast furnace stove is an integral part of the iron making process in the steel industry. These stoves are used to heat air which is then used in the blast furnace to chemically reduce iron ore to iron metal. The simulation of the stove's behavior is the first step in a program to reduce the cost of operating these stoves by minimizing the natural gas consumption during the heating cycle, while still maintaining a high enough output air temperature in the cooling cycle to drive the needed chemical reaction in the blast furnace. The formulation and solution of this optimal control problem will also be discussed. The solution technique used to solve the discrete representations of the model and control PDE's must be robust to linear systems with disparate eigenvalues, and must converge rapidly without using tuning parameters. The disparity in eigenvalues is created by the different time scales for convection in the gas, and conduction in the brick; combined with a difference between the scaling of the model and control PDE's. A preconditioned implicit Newton-Krylov solution technique was employed. The procedure employs Newton's method, where the update to the current solution at each stage is computed by solving a linear system. This linear system is obtained by linearizing the discrete approximation to the PDE's, using a numerical approximation for the Jacobian of the discretized system. This linear system is then solved for the needed update using a preconditioned Krylov subspace projection method.

Introduction

In this paper we discuss the use of an implicit Newton-Krylov method to solve a set of partial differential equations representing a physical model of a blast furnace stove. The blast furnace stove is an integral part of the iron making process in the steel industry. These stoves are used to heat air which is then used in the blast furnace to chemically reduce iron ore to iron metal. Internally these stoves consist of a combustion chamber and a large mass of refractory brick. The operation of the stoves is functionally divided into two phases. In the first phase (heating cycle), the brick is heated by burning a mixture of waste gas from the blast furnace itself and natural gas in the combustion chamber and allowing the hot exhaust gas to escape through holes (flues) in the brick. In the second phase (cooling cycle), normal air is heated by

*This work was performed under the auspices of the Department of Energy under contract W–7405–ENG–36.

forcing it through the flues in the heated brick and mixing the resulting hot air with ambient air in the combustion chamber to maintain a constant output temperature. This hot air is then used in the blast furnace to drive the desired chemical reaction. The simulation of the stove's behavior is the first step in a program to reduce the cost of operating these stoves by minimizing the natural gas consumption during the heating cycle, while still maintaining a high enough output air temperature in the cooling cycle to drive the needed chemical reaction in the blast furnace. The formulation and solution of this optimal control problem will also be discussed. A diagram of the stove is shown in Figure 1.

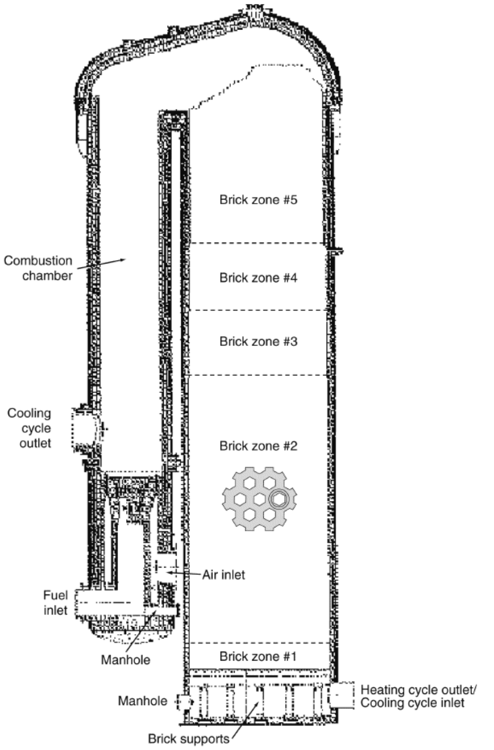


Figure 1: A diagram of a blast furnace stove. The chamber on the left is where combustion of the fuel gas takes place. The chamber on the right contains the refractory brick. The insert in Brick Zone #2 shows the shape of one of the bricks.

Model Description

Heat transfer within the stove is modeled with a set of transient partial differential equations (PDEs) that relate heat flux between the working gas and the storage brick. An energy equation describes the transient heat conduction within the storage brick, and the radiative transport between the working fluid and the storage medium. Specifically, the equations for the energy change in the gas and the brick during the heating cycle are

$$\rho_g\, \mathcal{C}_{p,g}\, \mathcal{A}_g \frac{\partial T_g}{\partial t} = h\, \mathcal{L}_g\, (T_s - T_g) - \mathcal{C}_{p,g} \frac{\partial T_g}{\partial z} \dot{m}_g(t) + Q_{comb}, \quad (1a)$$

$$\rho_s\, \mathcal{C}_{p,s}\, \mathcal{A}_s \frac{\partial T_s}{\partial t} = h\, \mathcal{L}_g\, (T_g - T_s)\,, \quad (1b)$$

$$\rho_g = \frac{pM_g}{\mathcal{R}T_g}, \quad (1c)$$

$$\Delta p = \frac{\mathcal{F}\, \dot{m}_g^2}{2\, \rho_g\, \mathcal{L}_g\, \mathcal{A}_g^2} \Delta z, \quad (1d)$$

$$\mathcal{C}_{p,g} = u(T_g) \quad h = v(T_g). \quad (1e)$$

In these equations $T_g(z,t)$ is the temperature of the gas, $T_s(z,t)$ is the temperature of the brick, and both vary over time t and space z. The densities of the gas and solid are, respectively, ρ_g and ρ_s. The heat capacity at constant pressure for the gas is $\mathcal{C}_{p,g}$, and the heat capacity at constant pressure for the brick is $\mathcal{C}_{p,s}$. The quantity $\mathcal{L}_g$ is the perimeter of a single hole in the brick. The quantity $\mathcal{A}_g$ is the area of one hole in the brick, and $\mathcal{A}_s$ is the area of the brick surrounding any one hole. The quantity h is the heat transfer coefficient between the gas and the brick, which consists of a portion due to convection in the gas and a portion due to radiation between the gas and the brick (*i.e.*, $h = h_{convection} + h_{radiation}$). The mass flow rate of gas through the stove is $\dot{m}_g(t)$. The molecular mass of the gas is M_g, its pressure is p, and $\mathcal{R}$ is the ideal gas constant. The friction factor for gas flow in the flues is $\mathcal{F}$.

Equation (1a) is the change in energy over time for the gas, while Equation (1b) is the energy change of the brick. The first term on the right hand side of Equation (1a) represents the *convection* of heat between the gas and the brick in the direction perpendicular to the flow of gas through the holes. The second term on the right of this equation represents the *convection* of heat in the gas in the direction parallel to the flow of gas in the holes. The right side of Equation (1b) represents the *convection* of heat between the solid and the gas, perpendicular to the gas flow. Note that the only quantity available for controlling the amount of heat in the stove is the mass flow rate $\dot{m}_g(t)$. The third term on the right hand side of Equation (1a) represents the heat added to the stove by continuing combustion of the fuel gas as it flows through the flues. The burners in these stoves were not designed to burn mixtures containing natural gas, and the operators believe that as a result, not all of the combustion takes place in the combustion chamber. So when the gas enters the flues some small percentage $\mathcal{U}$ of the fuel remains unburned. We assume that all of this remaining fuel is burned in the first $\mathcal{D}$ meters of each flue. As the gas travels down the flue for a distance dz, the heat from burning $\frac{\mathcal{U}}{\mathcal{D}}dz$ of the remaining fuel is added to the gas through the term Q_{comb}. This is continued until all of the fuel is consumed at distance $\mathcal{D}$.

Equations (1c), (1d), and (1e) describe the effect of gas temperature T_g on the density ρ_g, heat capacity $\mathcal{C}_{p,g}$, and heat transfer coefficient h. Accounting for this temperature variation is necessary because the gas temperatures vary between about 1500°C and 300°C. Equation (1c) is simply the ideal gas law, and Equation (1d) is an empirical correlation for the pressure drop in a pipe due to friction under turbulent flow conditions, which is discussed in Bird *et al.* (1960, page 188). The function that was used for $q(T_g)$ in Equation (1e) was the empirical correlation for heat transfer in a pipe under turbulent flow that is discussed in Bird *et al.* (1960, page 399). The function used for $p(T_g)$ was determined by fitting the data in Hilsenrath (1955) for the constituents of the exhaust gas.

Similarly the differential equations for the energy change in the gas and the brick during the cooling phase are

$$\rho_g \, \mathcal{C}_{p,g} \, \mathcal{A}_g \frac{\partial T_g}{\partial t} = h \, \mathcal{L}_g \, (T_s - T_g) - \mathcal{C}_{p,g} \frac{\partial T_g}{\partial z} \dot{m}_g(t), \tag{2a}$$

$$\rho_s \, \mathcal{C}_{p,s} \, \mathcal{A}_s \frac{\partial T_s}{\partial t} = h \, \mathcal{L}_g \, (T_g - T_s) \,, \tag{2b}$$

$$\dot{m}_g(t) = \left(1 - \frac{\mathcal{C}_{p,g} \, T_g(t) - \mathcal{C}_{p,g} \, T_{bo}}{\mathcal{C}_{p,g} \, T_g(t) - \mathcal{C}_{p,g} \, T_{bi}}\right) \dot{m}_{bi}, \tag{2c}$$

along with Equations (1c), (1d), and (1e). The air going into the blast furnace ideally must be maintained at a constant temperature T_{bo}. In order to achieve this goal, not all of the air is routed through the stove during the cooling phase. Rather, some of the air is diverted around the stove and is later mixed with the air heated by the stove to maintain the desired outlet temperature. The inlet air temperature T_{bi} is assumed to be constant. Since the temperature $T_g(t)$ of the air heated by the stove changes over time, the amount of air $\dot{m}_g(t)$ routed through the stove must also change over time. This change in the flow rate through the stove is defined by Equation (2c) and is referred to as the *bypass equation*. Normally, the quantity of air passing through the stove increases as the cooling phase progresses, due to the cooling of the bricks. The total mass flow rate into the stove–bypass system $\dot{m}_{bi}$ is assumed to be constant. A diagram depicting the mass and energy flow in the stove is shown in Figure 2. Note that the directions of both heat and mass flow during the cooling cycle are opposite those during the heating cycle.

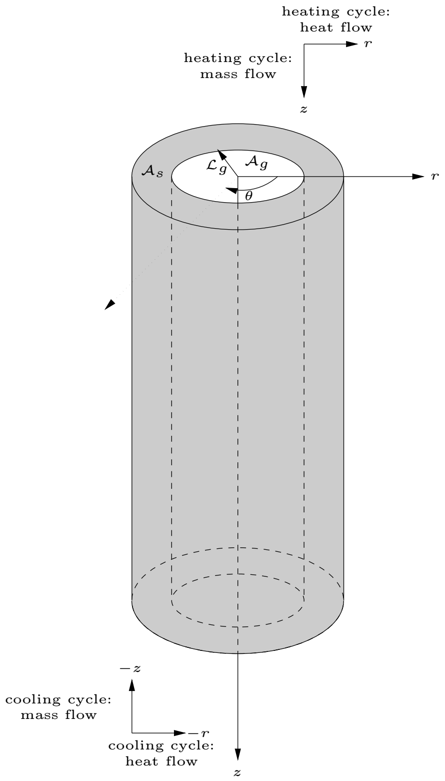


Figure 2: A diagram of the mass and energy flow in the stove for the two phases of its operation. The area of the white circle is $\mathcal{A}_g$, and the area of the gray annulus is $\mathcal{A}_s$. Note that a *cylindrical* coordinate system, centered at the middle of the hole in the brick, is used in this diagram.

For computational purposes Equations (1) and (2) must be discretely approximated. We use the finite-volume formulation to discretize these equations in both space and time. In this technique, the computational domain is divided into some number of non-overlapping control volumes such that only one grid point lies inside each control volume. This set of control volumes must completely cover the original domain. The differential equations are then integrated over each control volume. Denote the states of the system by the vector $\boldsymbol{x}^\dagger = [T_g \; T_s \; \rho_g \; \mathcal{C}_{p,g} \; h]$. These integrals are evaluated by approximating the variation of $\boldsymbol{x}$ between each grid point using piecewise profiles. This procedure results in a set of discrete equations containing values of $\boldsymbol{x}$ for each grid point. Intuitively these discrete equations define a conservation principle for $\boldsymbol{x}$ over the finite volume of each cell, just as the original differential equations express it for an infinitesimal volume. As an example consider discretizing the term $\mathcal{C}_p \frac{\partial T_g}{\partial z} \dot{m}_g(t) = 0$ Applying Gauss' theorem to the volume integral gives $\int_{\mathcal{V}} \mathcal{C}_p \, \dot{m}_g(t) \frac{\partial T_g}{\partial z} \, d\mathcal{V} = \mathcal{C}_p \int_{\mathcal{V}} \nabla \cdot (\dot{m}_g \, T_g) \, d\mathcal{V} = \mathcal{C}_p \oint_{\mathcal{S}} (\dot{m}_g \, T_g) \cdot \hat{n} \;\; d\mathcal{S} = 0$. In our problem, the dimensions of each control volume are identical, hence the discretized form of this term for the ith grid point is $\mathcal{C}_p \, \dot{m}_g(t) \, \dfrac{T_{g,i+1} - T_{g,i-1}}{2 \, \Delta z} = 0$. In this particular example the result is identical to that obtained with the finite-difference method, but in general this is not the case.

The optimal control problem is cast as a set of equations which minimize the area under the curve describing the mass flow rate $\dot{m}_g(t)$ over the entire heating cycle time. The logic behind this approach is that in order to minimize the cost of heating the stove, one must minimize the amount of fuel used to obtain the required total heat in the brick at the end of the heating cycle. So this optimization problem is a search over the space of all *functions* $\dot{m}_g(t)$, for the one which has minimum area under it, subject to the constraint that the amount of heat in the brick at the end of the heating cycle must exceed some specified value. There are numerous ways to formulate the search for such a function. One promising approach that we are pursuing involves framing the problem as finding the minimum of a scalar performance criterion function $J(T_g, T_s, \dot{m}_g(t))$ subject to a set of constraints $\boldsymbol{f}(T_g, T_s, \dot{m}_g(t)) = \boldsymbol{0}$, where the time-varying vector function $\boldsymbol{f}(\cdot)$ is given by rewriting Equations (1) and (2). One way to solve this problem is to "adjoin" the performance criterion to the constraint equations through a set of "undetermined multipliers" such that the resulting equation is $H = J(T_g, T_s, \dot{m}_g(t)) + \boldsymbol{p}_T^\dagger \, \boldsymbol{f}(T_g, T_s, \dot{m}_g(t))$. For this problem, intuitively the multipliers $\boldsymbol{p}_T$ represent the sensitivity of changes in the mass flow rate $\dot{m}_g(t)$ to changes in the temperatures T_g and T_s. An optimal mass flow rate satisfies the condition $\frac{\partial H}{\partial \dot{m}_g} = 0$. Carrying out the details of this formulation yields a set of equations for the time evolution of p_{T_g} and p_{T_s} analogous to those for T_g and T_s in Equations (1) and (2). This system of four partial differential equations can then be solved simultaneously for the optimal mass flow rate $\dot{m}_g(t)$.

Solution Technique

We chose an implicit Newton-Krylov technique to solve the discrete representations of Equations (1) and (2) because the solution technique must be robust for systems having disparate eigenvalues in the linear approximation, and it must provide rapid convergence without using tuning parameters. The disparity in eigenvalues is created by the different time scales for convection in the gas, and conduction in the brick. Rapid convergence is required in order to compute the necessary changes in the mass flow rate for the fuel gas during the heating cycle, with sufficient time to implement them as the heating cycle progresses. Lastly, a parameter-free method allows the use of the technique by steel company personnel with limited experience in non-linear solution techniques. The procedure employs Newton's method, where the update to the current solution at each stage is computed by solving a linear system. This linear system results from linearizing the discrete approximation to the PDE's, using a numerical approximation for the Jacobian of the discretized system. This linear system is then solved for the needed update using a preconditioned Krylov subspace projection method.

Note that the system defined by Equations (1) and (2) consists of *both* differential and algebraic equations. The solutions consist of functions of both the distance z and the time t. The time variation of the functions T_g and T_s must be tracked because in both the heating and cooling cycles the system does *not* come to steady state. This means that at the end of either cycle the time derivatives $\frac{\partial T_g}{\partial t}$ and $\frac{\partial T_s}{\partial t}$ are not small. Because the solution that must be tracked varies over space and time, a two-dimensional grid was used to discretize Equations (1) and (2). The conditions at the initial time become conditions along a boundary of the time dimension of this grid, in analogy to elliptic problems. This treatment of time allows the temporal derivatives to be incorporated into the Jacobian of the discrete system in the same manner that spatial derivatives are normally incorporated.

As indicated above, our solution technique breaks down naturally into two parts. The first part consists of searching for a nonlinear update to the current solution. Conceptually, Equations (1) and (2) can be rewritten as the vector equation $\boldsymbol{f}_{\mathrm{c}}(T_g, T_s, \rho_g, \mathcal{C}_{p,g}, h) = \boldsymbol{0}$. An approximate solution to this differential algebraic system is given by a set of states T_g, T_s, ρ_g, $\mathcal{C}_{p,g}$, and h which make the value of $\boldsymbol{f}_{\mathrm{c}}(\cdot)$ close to zero. Intuitively, this is a root finding problem in which the roots are *functions* of the distance z and the time t. The discretized version of $\boldsymbol{f}_{\mathrm{c}}(\cdot)$ is called the *nonlinear residual* and is denoted $\boldsymbol{f}_{\mathrm{d}}(\cdot)$. Collectively the states are denoted by the vector $\boldsymbol{x}^{\dagger} = [T_g\, T_s\, \rho_g\, \mathcal{C}_{p,g}\, h]$. The root finding problem is to find the state $\boldsymbol{x}$ which minimizes the nonlinear residual $\boldsymbol{f}_{\mathrm{d}}(\boldsymbol{x})$. One way to solve this problem is to compute the second order Taylor series expansion of $\boldsymbol{f}_{\mathrm{d}}(\boldsymbol{x})$ about the point $\boldsymbol{x}$

$$\begin{aligned} f_{\mathrm{d},i}(\boldsymbol{x}+\delta\boldsymbol{x}) = f_{\mathrm{d},i}(\boldsymbol{x}) + \sum_{j=1}^{\mathfrak{n}} \frac{\partial f_{\mathrm{d},j}}{\partial x_j}\delta x_j \\ + O(\delta\boldsymbol{x}^2). \end{aligned} \tag{3}$$

Neglecting terms of order $\delta\boldsymbol{x}^2$ and higher and setting $\boldsymbol{f}_{\mathrm{d}}(\boldsymbol{x}+\delta\boldsymbol{x}) = \boldsymbol{0}$, we obtain a set of linear equations for the corrections $\delta\boldsymbol{x}$ that move each residual toward zero simultaneously. For the kth iteration of the algorithm, the vector form of these equations is

$$\boldsymbol{J}_{\boldsymbol{f}}(\boldsymbol{x}_k)\,\delta\boldsymbol{x}_k = -\boldsymbol{f}_{\mathrm{d}}(\boldsymbol{x}_k), \tag{4}$$

where $\boldsymbol{J}_{\boldsymbol{f}}(\boldsymbol{x}_k)$ is the Jacobian matrix of the discrete system $\frac{\partial \boldsymbol{f}_{\mathrm{d},k}}{\partial \boldsymbol{x}_k}$. The corrections are added to the solution vector giving the update rule

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k\,\delta\boldsymbol{x}_k, \tag{5}$$

where $\alpha_k \in (0,1]$ is a weighting factor to keep the algorithm from overshooting the solution. This algorithm for root solving is commonly known as the *Newton-Raphson method* or simply *Newton's method*, and it is discussed in Fletcher (1987).

The second part of the algorithm consists of finding the solution for the linear system in Equation (4). This equation is of the general form $\boldsymbol{A}\,\boldsymbol{y} = \boldsymbol{b}$, where $\boldsymbol{A}$ is an $(\mathfrak{n} \times \mathfrak{n})$ matrix. The method that we use is a conjugate-gradient-like polynomial-based iterative scheme. The general solution update is

$$\begin{aligned} \boldsymbol{y}_l = \boldsymbol{y}_0 + (\gamma_{l0}\,\boldsymbol{r}_0 + \gamma_{l1}\,\boldsymbol{A}\,\boldsymbol{r}_0 + \gamma_{l2}\,\boldsymbol{A}^2\,\boldsymbol{r}_0 \\ + \cdots + \gamma_{l(l-1)}\,\boldsymbol{A}^{l-1}\,\boldsymbol{r}_0), \end{aligned} \tag{6}$$

where $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\,\boldsymbol{y}_0$, $\boldsymbol{r}_l$ is the linear residual at step l, and $\boldsymbol{y}_0$ is the initial guess for the solution of the linear system. This means that the solution $\boldsymbol{y}_l$ at step l is the initial solution $\boldsymbol{y}_0$ plus a linear combination of vectors in the set $\{\boldsymbol{r}_0,\ \boldsymbol{A}\,\boldsymbol{r}_0,\ \boldsymbol{A}^2\,\boldsymbol{r}_0,\ \ldots,\ \boldsymbol{A}^{l-1}\,\boldsymbol{r}_0\}$. The space spanned by this set of vectors is the *Krylov subspace*, which is denoted by $\mathcal{K}_l(\boldsymbol{r}_0,\,\boldsymbol{A})$. Since new solution approximations are computed by projecting the linear residual $\boldsymbol{r}_l$ onto a Krylov subspace, these algorithms are collectively known as Krylov subspace projection methods.

Equation (6) can be written in the simpler form $\boldsymbol{y}_l = \boldsymbol{y}_0 + \sum_{j=0}^{l} \gamma_{lj}\,\boldsymbol{p}_j$. The manner in which $\boldsymbol{p}_l$ is computed defines a particular Krylov subspace method. In general, two criteria can be used to compute the $\boldsymbol{p}_l$ vectors. The first criterion is to pick $\boldsymbol{p}_l$ to minimize some norm of the current linear residual $\boldsymbol{r}_l$. The second criterion is to choose $\boldsymbol{p}_l$ so that the the current linear residual $\boldsymbol{r}_l$ is orthogonal to some set of vectors $\mathcal{L}_l$, where $\mathcal{L}_l$ may be different from $\mathcal{K}_l$. Mathematically, these two criteria are

$$\min_{p_l \in \mathcal{K}_l} \|\boldsymbol{r}_l\|_{\mathcal{N}} = \min_{p_l \in \mathcal{K}_l} \left\| \boldsymbol{r}_0 - \boldsymbol{A} \sum_{j=0}^{l} \gamma_{lj}\,\boldsymbol{p}_j \right\|_{\mathcal{N}}, \tag{7}$$

$$\boldsymbol{r}_l = \left(\boldsymbol{r}_0 - \boldsymbol{A} \sum_{j=0}^{l} \gamma_{lj}\,\boldsymbol{p}_j \right) \perp \mathcal{L}_l, \tag{8}$$

where $\|\cdot\|_{\mathcal{N}}$ represents an arbitrary norm. By satisfying the first criterion, the algorithm is guaranteed to converge to a solution which minimizes some measure of the error between the exact and approximate solutions. By satisfying the second criterion, the algorithm is guaranteed to converge in a finite number of iterations. The conjugate gradient algorithm is derived assuming that $\boldsymbol{A}$ is symmetric positive definite, in which case both of these criteria can be satisfied with the same algorithm, for $\mathcal{L}_l \equiv \mathcal{K}_l$.

In most cases the Jacobian is *not* symmetric positive definite, hence both of the above criteria can not be satisfied simultaneously. There are numerous algorithms based on different implementations of one of these two criteria. The technique that we use is the Generalized Minimal Residual (GMRES) algorithm developed by Saad and Schultz (1986). This algorithm has three distinguishing features. First, it is guaranteed to minimize the 2-norm of the linear residual $\|\boldsymbol{r}_l\|_2 = \left\|\boldsymbol{r}_0 - \boldsymbol{A}\sum_{j=0}^{l}\gamma_{lj}\,\boldsymbol{p}_j\right\|_2 = \|\boldsymbol{b} - \boldsymbol{A}\,\boldsymbol{y}_l\|_2$. Second, the search directions $\boldsymbol{p}_l$ are $\boldsymbol{I}$-orthonormal, meaning that $\boldsymbol{p}_i^{\dagger}\,\boldsymbol{p}_j = 0$ for all $i \neq j$, and $\|\boldsymbol{p}_i\|_2 = 1$. Third, the linear residual at any iteration is $\boldsymbol{A}$-orthogonal to all previous search directions, meaning $\boldsymbol{r}_i^{\dagger}\boldsymbol{A}\,\boldsymbol{p}_j = 0$ for all $i > j$. Another way to state the last condition is that the linear residual $\boldsymbol{r}_l$ is orthogonal to the Krylov subspace $\mathcal{L}_l = \boldsymbol{A}\,\mathcal{K}_l(\boldsymbol{r}_0,\,\boldsymbol{A})$.

The speed of convergence for finding the solution

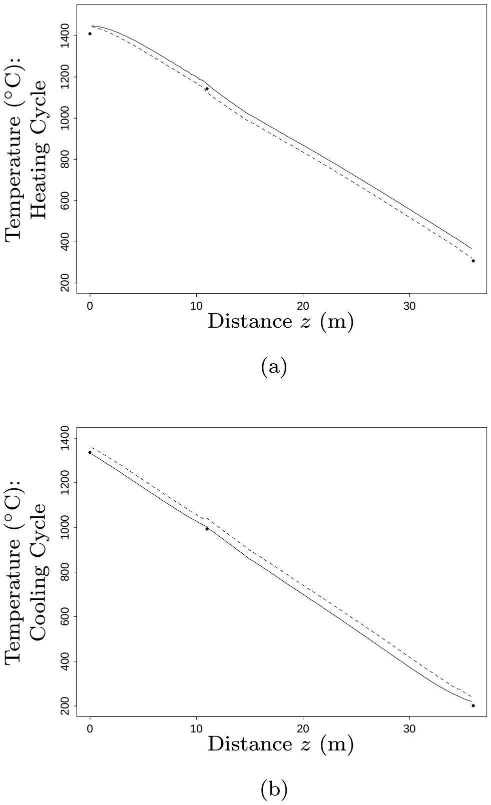
$\boldsymbol{y}$ of the linear system $\boldsymbol{A}\,\boldsymbol{y} = \boldsymbol{b}$ depends on the ratio of the maximum to the minimum eigenvalues of the matrix $\boldsymbol{A}$. If this ratio is large, then $\boldsymbol{A}$ is said to be poorly conditioned. In many practical cases $\boldsymbol{A}$ is so poorly conditioned that Krylov methods, such as GMRES, do not converge at all. Preconditioning makes a linear system easier to solve by improving the condition of the $\boldsymbol{A}$ matrix. Preconditioning is accomplished by multiplying both sides of the linear system by a matrix $\boldsymbol{P}$ which resembles $\boldsymbol{A}^{-1}$ in some sense. The new system $\boldsymbol{P}\,\boldsymbol{A}\,\boldsymbol{y} = \boldsymbol{P}\,\boldsymbol{b}$ is now easier to solve than the original system. In our application the preconditioner is applied to Equation (4) prior to solving for the linear correction $\delta\boldsymbol{x}_k$ using GMRES. Our preconditioner is an incomplete LU factorization of the Jacobian matrix $\boldsymbol{J}_{\boldsymbol{f}}(\boldsymbol{x}_k)$ with level of fill-in $\mathbb{m}$ (*i.e.*, ILU($\mathbb{m}$)).

This technique has numerous positive features. It has been proven that the upper bound on the convergence rate of Newton's method is quadratic, and the bound for conjugate-gradient methods is superlinear. This means that our solution method is capable of very fast convergence. Since the algorithm is implicit we can follow any time scale in the problem, rather than being forced to follow the fastest time scale, as in explicit methods. This algorithm directly minimizes both the absolute and relative error of the solution. Because this method is based on root finding, the resulting solution is one for which $\boldsymbol{f}_{\mathrm{d}}(\boldsymbol{x}_k) \approx \mathbf{0}$, and $\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|_{\mathcal{N}} \approx 0$ for some iteration k. Also this algorithm has modest memory requirements, and has very few, if any, parameters.

Of course this algorithm also has some negative features. In practice Newton's method often diverges unless it is started fairly close to a root. Furthermore, for roots with order greater than one, the upper bound on the convergence rate is linear. In this application, although both of these difficulties are still possible, there is one feature of the problem which simplifies matters. For a well-posed system of differential equations, there is a unique real-valued solution which depends continuously on the initial and boundary conditions. *Assuming* that this is also true for the discretized system, there is only *one real root* for $\boldsymbol{f}_{\mathrm{d}}(\boldsymbol{x}_k) = \mathbf{0}$. The fact that there is only one real root may simplify the task of computing it. Newton's method can be made more robust to the initial guess by adjusting the size of the Newton step taken in each iteration k using the parameter α_k in Equation (5). Some practical methods for doing this are discussed in Press *et al.* (1992, pages 383–393). Another potential difficulty is that GMRES is not guaranteed to converge in a finite number of iterations. This difficulty is dealt with by preconditioning the linear system in Equation (4). The goal of preconditioning is to make this equation much easier to solve without expending much computational effort constructing the preconditioner.

Simulation Results

First we will discuss the agreement between our model and data taken from the stoves. Using the model, it is possible to compute a temperature profile down the length of the stove for both the gas and the brick. Two plots showing the temperature profiles in both the gas and the brick are shown in Figure 3. Figure 3(a) shows the temperature versus distance at the end of a heating cycle. The solid line is the gas temperature and the dashed line is



(a)

(b)

Figure 3: (a) The temperature versus distance at the end of a heating cycle. The solid line is the gas temperature and the dashed line is the brick temperature. The three points ● are temperature measurements taken in the actual stove at these distances.
(b) The temperature versus distance at the end of a cooling cycle.

the brick temperature. The three points ● are temperature measurements taken in the actual stove at these distances. Figure 3(b) shows the temperature versus distance at the end of a cooling cycle. Note that the zero distance in Figure 3 is referenced to the top of Brick zone #5 in Figure 1. Note that there appears to be reasonable agreement between our model and the actual data. These plots also illustrate that at the end of both cycles, the system is *not* at thermal equilibrium. If thermal equilibrium were reached, the gas and brick temperatures would be identical and both would be constant over the length of the stove. Notice that there are only three temperature measurements being made down the length of the actual stoves. The biggest problem in assessing the validity of the model is this lack of measurements from the actual system.

One measure of the state of the stove for which more data is available is the amount of air *not* sent through the stove during the cooling cycles. This is usually expressed as the percentage of the total volumetric flow rate into the stove-bypass system that is actually routed through the bypass. Measurements are made every 15 seconds of the total flow rate into the stove-bypass system, and of the flow rate into the stove. The percentage sent through the bypass can easily be computed from these two measurements. A comparison of the values computed by our model and the actual data is shown in Figure 4. Note that time is measured relative to the total time for a combination of one heating cycle and one cooling cycle. Four such complete cycles are shown in the figure.

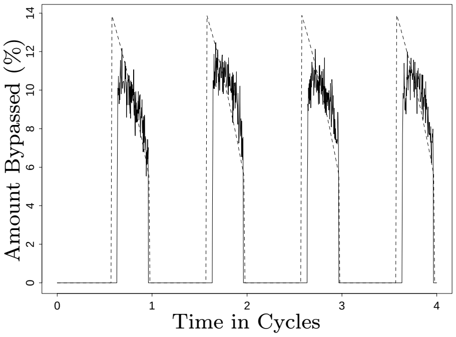
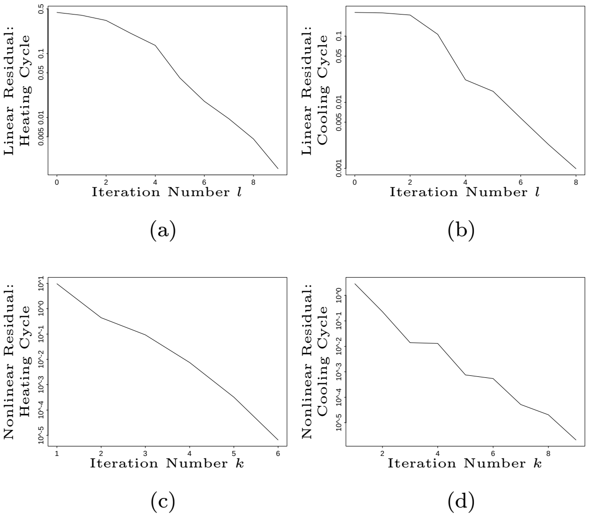


Figure 4: The percentage of the total volumetric flow rate that is routed through the bypass as a function of time.

Next we will discuss the convergence properties of our algorithm with respect to the number of iterations. As discussed in Luenberger (1984), the rate of convergence of a sequence $\{\boldsymbol{u}_n\}_{n=0}^{\infty}$ which converges to a limit $\boldsymbol{u}^*$ is assessed by computing $\beta = \lim_{n\to\infty} \frac{\|\boldsymbol{u}_{n+1}-\boldsymbol{u}^*\|}{\|\boldsymbol{u}_n-\boldsymbol{u}^*\|^{\mathrm{p}}}$ where p is a positive integer. The *order of convergence* is the largest number p for which $0 \le \beta < \infty$. If $\mathrm{p} = 1$ and $0 < \beta < 1$, then the convergence rate is said to be linear. If $\mathrm{p} = 1$ and $\beta = 0$, then the convergence rate is superlinear. If $\mathrm{p} = 2$, then the convergence rate is quadratic. For example, given a real number a such that $0 < a < 1$, the sequence $u_n = a^n$ converges linearly, the sequence $u_n = a^{n^2}$ converges superlinearly, and $u_n = a^{2^n}$ converges quadratically. The convergence properties of our Newton-Krylov algorithm are illustrated in Figure 5. Figures 5(a) and 5(b) plot the linear residual $\|\boldsymbol{f}_{\mathrm{d}}(\boldsymbol{x}_k) + \boldsymbol{J}_{\boldsymbol{f}}(\boldsymbol{x}_k)\,\delta\boldsymbol{x}_l\|_2$ versus the linear iteration number l for a fixed nonlinear iteration k during a heating cycle and a cooling cycle respectively. These two figures show the worst case convergence of ILU(m) preconditioned GMRES while solving for $\delta\boldsymbol{x}_k$ in Equation (4). Similarly, Figures 5(c) and 5(d) plot the nonlinear residual $\|\boldsymbol{f}_{\mathrm{d}}(\boldsymbol{x}_k)\|_2$ versus the nonlinear iteration k during a heating cycle and a cooling cycle respectively. These two figures show the worst case convergence of Newton's



(a)

(b)

(c)

(d)

Figure 5: (a) The worst case convergence of ILU(m) preconditioned GMRES while solving for $\delta\boldsymbol{x}_k$ in Equation (4) during a heating cycle.
(b) The worst case convergence of ILU(m) preconditioned GMRES while solving for $\delta\boldsymbol{x}_k$ in Equation (4) during a cooling cycle.
(c) The worst case convergence of Newton's method while solving for $\boldsymbol{x}_{last}$ using Equation (5) during a heating cycle.
(d) The worst case convergence of Newton's method while solving for $\boldsymbol{x}_{last}$ using Equation (5) during a cooling cycle.

method while solving for the final solution $\boldsymbol{x}_{last}$ using Equation (5). Note that the vertical axes of these plots are logarithmic. Intuitively, on a semi-logarithmic plot, a sequence which converges linearly will appear as a straight line. A sequence converging faster than linearly will have a negative curvature (*i.e.*, curving downward) and one converging slower than linearly will have positive curvature (*i.e.*, curving upward). Therefore Figures 5(a) and 5(b) indicate that ILU($\mathfrak{m}$) preconditioned GMRES is converging superlinearly. Figures 5(c) and 5(d) seem to indicate that Newton's method is converging linearly. It might be argued from these plots that we are achieving a slow superlinear convergence with Newton's method, but in any case it seems clear that quadratic convergence is not being obtained.

We also investigated convergence with respect to cell volume in the discrete approximation. Intuitively, as the cell volume is decreased at some point the changes in the simulation results should become very small. We performed a grid convergence study and used the coarsest grid for which the solution became invariant to the grid size. Specifically we use 100 cells over the 36 meter length of the stove, and 40 cells over the 90 minute single cycle time. This means that the results in this section are for grid converged solutions in both space and time.

Conclusion

In this paper we have presented an implicit Newton-Krylov method which we have used to simulate a physical model of a blast furnace stove. The simulation of the stove's behavior is the first step in a program to reduce the cost of operating these stoves by minimizing the natural gas consumption during the heating cycle, while still maintaining a high enough output air temperature in the cooling cycle to drive the needed chemical reaction in the blast furnace. The Newton-Krylov technique was selected for several reasons. It is robust for solving systems having components which evolve at very different time scales. In this application, this problem is particularly acute during the cooling cycle wherein the time scale of the bypass computation in Equation (2c) is much faster than the time scale of the gas heating in Equation (2a), which is in turn much faster than the time scale of the brick cooling in Equation (2b). The algorithm converges rapidly to a solution, which is necessary in order to compute near real-time changes in the fuel gas flow rate as the heating cycle progresses. The method is also parameter-free which is needed because the stove operators have no experience with non-linear differential equation solvers.

References

Bird, R., Stewart, W., & Lightfoot, E. (1960). *Transport Phenomena.* John Wiley & Sons, Inc., New York, NY.

Fletcher, R. (1987). *Practical Methods of Optimaization* (2nd edition). John Wiley & Sons, Ltd., Chichester, United Kingdom.

Hilsenrath, J. (1955). *Tables of Thermal Properties of Gases.* No. 564 in National Bureau of Standards Circular. U.S. Government Printing Office, Washington, DC.

Luenberger, D. (1984). *Linear and Nonlinear Programming* (2nd edition). Addison-Wesley Publishing Co., Inc., Reading, MA.

Press, W., Teukolsky, S., Vetterling, W., & Flannery, B. (1992). *Numerical Recipes in C: The Art of Scientific Programming* (2nd edition). Cambridge University Press, Cambridge, United Kingdom.

Saad, Y., & Schultz, M. (1986). GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM Journal of Scientific and Statistical Computing*, *7*(3), 856–869.